Computer Modeling in
Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2023.030535

ARTICLE

Fractional Gradient Descent RBFNN for Active Fault-Tolerant Control of Plant Protection UAVs

Lianghao Hua[1,2], Jianfeng Zhang[1,*], Dejie Li[3] and Xiaobo Xi[1]

[1]School of Mechanical Engineering, Yangzhou University, Yangzhou, 225009, China

[2]College of Intelligent Manufacturing, Yangzhou Polytechnic Institute, Yangzhou, 225009, China

[3]School of Automation, Nanjing University of Aeronautics and Astronautics, Nanjing, 211106, China

*Corresponding Author: Jianfeng Zhang. Email: zhangjf@yzu.edu.cn

Received: 12 April 2023 Accepted: 26 July 2023 Published: 15 December 2023

ABSTRACT

With the increasing prevalence of high-order systems in engineering applications, these systems often exhibit significant disturbances and can be challenging to model accurately. As a result, the active disturbance rejection controller (ADRC) has been widely applied in various fields. However, in controlling plant protection unmanned aerial vehicles (UAVs), which are typically large and subject to significant disturbances, load disturbances and the possibility of multiple actuator faults during pesticide spraying pose significant challenges. To address these issues, this paper proposes a novel fault-tolerant control method that combines a radial basis function neural network (RBFNN) with a second-order ADRC and leverages a fractional gradient descent (FGD) algorithm. We integrate the plant protection UAV model's uncertain parameters, load disturbance parameters, and actuator fault parameters and utilize the RBFNN for system parameter identification. The resulting ADRC exhibits load disturbance suppression and fault tolerance capabilities, and our proposed active fault-tolerant control law has Lyapunov stability implications. Experimental results obtained using a multi-rotor fault-tolerant test platform demonstrate that the proposed method outperforms other control strategies regarding load disturbance suppression and fault-tolerant performance.

KEYWORDS

Radial basis function neural network; plant protection unmanned aerial vehicle; active disturbance rejection controller; fractional gradient descent algorithm

1 Introduction

In the past decade, multi-rotor unmanned aerial vehicle (UAV) technology has developed rapidly and is widely used in various industries, such as national defense, construction, power, rescue, and agriculture [1–5]. Different types of components, such as onboard computers, sensors, cameras, and controllers, are carried by multirotor UAVs to meet the requirements of various mission scenarios. Furthermore, multi-rotor UAVs' external disturbance rejection capability and fault tolerance also vary according to different scenarios. Therefore, many researchers have studied flight control algorithms for specific types of UAVs and their application scenarios [6–8].

The control system of a multi-rotor UAV is a typical underactuated system with nonlinear characteristics [9,10]. In the control algorithm of multirotor UAVs, the most common methods are the PID control algorithm [11], sliding mode control algorithm [12–14], and robust control algorithms such as disturbance rejection control [6,15,16]. With the increasing complexity of multirotor UAVs [17–20], researchers have considered combining adaptive control technology with robust control technology to better control such complex systems [14]. To address the uncertain dynamic parameters and external disturbances in the system, Lu et al. used neural networks to identify uncertain parameters in the system, which performed well in the control of such nonlinear systems [21]. However, the instability of the control system due to external or self-disturbances, as well as actuator and sensor failures, is a common phenomenon for UAVs [22,23]. Therefore, many researchers have considered combining intelligent control algorithms with robust and adaptive control to propose fault-tolerant control algorithms that can solve a class of faults [24–26]. Boche et al. proposed a fault-tolerant control scheme that utilizes continuous and discrete methods to address multiple actuator failures [27]. Alternatively, Wen et al. proposed an adaptive neural fuzzy sliding mode control method with online parameter updating in reference [28] to solve the fault-tolerant control problem of uncertain nonlinear systems subjected to actuator effectiveness failures and input saturation, regardless of fault limits. However, in some fault-tolerant control methods, the fuzzy neural network structure remains fixed and cannot independently adjust the fuzzy rule number, leading to increased computational burden and potential negative impacts on control performance when inputs and outputs increase. To address this issue, Yu et al. proposed an actuator fault-tolerant control method that utilizes a self-constructed fuzzy neural network and multivariable sliding mode control [23]. By combining adaptive algorithms with self-constructed fuzzy neural networks, the proposed method approximates actuator fault information and model uncertainty parameters, thus reducing the computational burden of the control system and ensuring stable fault systems within a finite time. Moreover, radial basis functions offer significant advantages in dealing with parameter approximation of nonlinear systems. Similarly, Xiang et al. used an observer combined with RBFNN to detect unknown faults in the system with changing model dynamics in the attitude system of the aircraft. Because of the advantages of neural networks in function approximation and data classification processing, they can handle model uncertainty changes and distinguish fault data in the case of actuator failures [29]. However, the fault-tolerant control method implemented has certain requirements for the construction of the system model. Therefore, it is necessary to find a control algorithm that does not overly rely on the system model. Zhou et al. proposed a fault-tolerant control method combining self-disturbance control and a radial basis function neural network to give the system a certain fault-tolerant control ability for actuator and sensor failures while resisting external disturbances [15]. The active disturbance rejection control algorithm has the characteristic of not overly relying on the system model, and is a good solution for control systems with uncertain models. However, for systems where changes in system model parameters are detected, using machine learning algorithms is also a good solution [30]. Zhong et al. introduced the model reference adaptive control into the RBFNN for online identification of the system model's changing parameters [16]. Similarly, Liu et al. designed a new cascade double loop ADRC method of neural network-based extended state observer (NNESO) to solve the attitude control problem of hypersonic vehicles in order to solve the disturbance problem of hypersonic vehicles [22]. The controller design ideas in this article have given us some inspiration. To better identify the uncertain items of dynamic parameters of systems, Hua et al. used optimized parameters of a self-disturbance control algorithm and combined them with a spatiotemporal RBFNN to achieve fault-tolerant control for actuator failures in a class of nonlinear systems, and significant results were obtained [31]. This article further studies on this basis.

However, the above research needs further investigation into the disturbance and fault-tolerant control algorithms for plant protection UAVs under heavy loads. Moreover, with the rapid development and application of plant protection UAVs, research on safety control mechanisms for this type of UAV has become increasingly important [32]. For this reason, the disturbance problem of UAV attitude control caused by loads in various load situations has received attention from many researchers [33]. Guerrero-Sanchez et al. proposed a hybrid control method combining a feedback linearization controller and artificial neural network for a UAV's suspended load situation, using the artificial neural network as an estimator of load disturbance to achieve disturbance control of the UAV under the load situation [34]. Similarly, Ameya et al. proposed a self-attenuation control strategy based on an extended state observer for effective payload mathematical modeling of UAVs. The approach considers the payload as an external disturbance of the system, employs the extended state observer to estimate the disturbance term, and then designs a self-attenuation controller to achieve stable control of the UAV [8].

In summary, plant protection UAVs are prone to actuator failures and must carry a significant amount of pesticide, thus requiring solutions for disturbance suppression and fault-tolerant control. To effectively control the UAV while carrying pesticides, we build upon the second-order ADRC developed by Herbst [35] and design an extended state observer (ESO). Additionally, to achieve fault-tolerant flight control during actuator failure, we propose a novel approach combining a radial basis function neural network (RBFNN) with a fractional-order gradient descent algorithm-based extended state observer to identify system generalized parameter terms [36]. This method utilizes a convex combination of the traditional RBFNN and a fractional gradient descent method with improved Riemann-Liouville derivatives, resulting in better nonlinear system identification capabilities than traditional RBFNN. We compared and analyzed the proposed fault-tolerant controller with traditional RBFNN and NNESO based ADRC controllers to evaluate the performance of fault-tolerant flight control. Through simulations, we have observed that the proposed method exhibits smaller overshoots for limited actuator gain and bias failures and efficiently suppresses oscillations caused by load disturbances. Our research contributions can be summarized as follows:

1. A second-order ADRC was designed to estimate nonlinear systems' uncertain parameters and load disturbances, which can suppress load disturbances and the impact of uncertain system modeling parameters with minimal parameter adjustment. The experimental results showed that this method could better suppress load disturbances and faster response speed.

2. An optimized extension state observer was proposed based on the second-order ADRC and a novel RBFNN with a fractional gradient descent algorithm with the optimal system identification solution. The output of the gradient descent RBFNN was combined with the high-order parameter estimation part of the ADRC observer to enhance the UAV's ability in disturbance suppression and reduce the error between the aircraft's attitude control angle and the desired angle.

3. The actuator's bias and gain fault items were integrated into the general parameter term, and the number of design parameters for the adaptive control was optimized. Flight state data of a plant protection UAV with actuator faults and loads were collected for model training of the gradient descent RBFNN, enabling the ADRC to have anti-disturbance and fault-tolerant control capabilities for the UAV in the presence of actuator faults and disturbances.

This article describes an independent fault-tolerant flight control design for a quadrotor aircraft's pitch, roll, and yaw subsystems. The proposed approach utilizes the aircraft's attitude angles and angular velocities to train weight parameters of a fractional-order RBFNN, which can better approximate the values of the fused parameter terms in the presence of modeling error, uncertainty, disturbance, and fault. Moreover, the proposed fault-tolerant flight control scheme can simultaneously handle multiple limited faults in the actuators. The stability of the designed control law is proven using Lyapunov stability analysis. Finally, simulation experiments are conducted on a fault injection platform for plant protection UAVs to demonstrate the excellent performance of the proposed algorithm.

The rest of this paper is organized as follows. Section 2 presents the aerodynamic model of the plant protection UAV, including the load model and actuator fault model. Section 3 introduces the design of a second-order ADRC with a generalized observer and an RBFNN based on the FGD learning algorithm. In Section 4, a second-order ADRC fault-tolerant controller with Lyapunov stability is designed based on the backstepping control approach. Section 5 describes the fault injection simulation experiments conducted on the Qball-X4 UAV using the modified model parameters that match the plant protection UAV and evaluates the performance of the designed fault-tolerant controller. Finally, Section 6 summarizes the work presented in this paper and discusses potential future research directions.

2 Mathematical Model of X-Type Quadrotor UAV

This section introduces the dynamic model of a quadrotor aircraft carrying a pesticide load. The aircraft's center of mass changes continuously during the flight due to the absence of a fixed form of pesticides and other spraying liquids in the water tank of the plant protection UAV. To simplify the analysis of the dynamic model of the quadrotor aircraft, we make the following assumptions:

1. The centroid of the water tank body remains unchanged.
2. Approximating a pesticide liquid in an aqueous phase as a particle.
3. Neglecting aerodynamic effects on aircraft and mounted water tanks.
4. Water tank and aircraft as a completely rigid structure.

To develop an effective control algorithm for quadrotor aircraft, it is imperative to obtain accurate state information regarding the aircraft's angular orientation, angular acceleration, linear velocity, and linear acceleration. As such, this section provides a brief analysis of the interrelationship between the airframe coordinate system and the inertial coordinate system of the quadrotor UAV, as well as an examination of the mathematical expressions of the parameters pertinent to quadrotor dynamics.

2.1 Quadrotor Dynamics Model

The dynamics model of a quadrotor aircraft has been proposed in many papers on quadrotor aircraft control [9,10]. To obtain the mathematical model of the plant protection UAV with a liquid load, we present in Fig. 1 the kinematic parameters of a quadrotor aircraft with a liquid load in a water tank.

Moreover, we established the body coordinate system $O_B - x_b y_b z_b$ (body frame) and the inertial coordinate system $O_I - x_i y_i z_i$ (inertia frame) based on the X-shaped structure framework of the quadrotor. In this section, the curved motion of the Earth's rotation and center of gravity are neglected when establishing the coordinate system. The generalized coordinates P_i and generalized velocities $\dot{P}_i$ are given by the following equations, respectively, $q = \begin{bmatrix} x & y & z & \theta_l & \phi_l & \phi & \theta & \psi \end{bmatrix}^T$ and $\dot{q} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} & \dot{\theta}_l & \dot{\phi}_l & \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T$. $X_B = \begin{bmatrix} x & y & z \end{bmatrix}^T$ is defined as the position of the quadrotor aircraft in the inertial frame. $\begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T$ is defined as the aircraft's corresponding Euler angles (roll, pitch, and yaw). Among them, m is the mass of the quadrotor aircraft (including the mass of the empty water tank), and l is the axis distance of the aircraft. The input matrix of the system is assumed to be

$$U = \begin{bmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \\ u_4(t) \end{bmatrix} = \begin{bmatrix} F_{\mathrm{T}} \\ F_2 + F_3 - F_1 - F_4 \\ F_1 + F_3 - F_2 - F_4 \\ T_{yaw,1} + T_{yaw,2} - T_{yaw,3} - T_{yaw,4} \end{bmatrix} = \begin{bmatrix} -\sum_{i=1}^{4} c_{T,i} \cdot \Omega_i^2 \\ c_{T,2}\Omega_2^2 + c_{T,3}\Omega_3^2 - c_{T,1}\Omega_1^2 - c_{T,4}\Omega_4^2 \\ c_{T,1}\Omega_1^2 + c_{T,3}\Omega_3^2 - c_{T,2}\Omega_2^2 - c_{T,4}\Omega_4^2 \\ c_{y,1}\Omega_1^2 + c_{y,2}\Omega_2^2 - c_{y,3}\Omega_3^2 - c_{y,4}\Omega_4^2 \end{bmatrix} \tag{1}$$

where F_T is the total tension of the four motors, the tension produced by each motor (N) is $F_i = c_{T,i} \cdot \Omega_i^2$, parameter $\Omega_i(r/\min)$ is the rotor speed (subscript $i = 1, 2, 3, 4$ is the corresponding rotor of the quadrotor aircraft), $c_{T,i}$ is the comprehensive propeller tension coefficient, $T_{yaw,i}$ is the torque produced ($N \cdot m$) by a single motor in the yaw direction, $c_{y,i}$ is the propeller torque coefficient, $u_1(t)$ is the throttle control amount of the quadrotor aircraft, which is the control amount of the altitude control subsystem, $u_2(t)$ is the input control amount of the roll subsystem, $u_3(t)$ is the input control amount of the pitch subsystem, and $u_4(t)$ is the input control amount of the yaw subsystem. Through reference to the literature [9,10], we obtained the general expressions for the nonlinear attitude dynamics model and position dynamics model of the quadrotor aircraft.

$$\begin{cases} \ddot{\phi} = \dfrac{\dot{\theta}\dot{\psi}\left(J_{zz} - J_{yy}\right)}{J_{xx}} - \dfrac{\dot{\theta}\Omega_g J_{RP}}{J_{xx}} + \dfrac{l}{J_{xx}} u_2(t) \\ \ddot{\theta} = \dfrac{\dot{\phi}\dot{\psi}\left(J_{xx} - J_{zz}\right)}{J_{yy}} + \dfrac{\dot{\phi}\Omega_g J_{RP}}{J_{yy}} + \dfrac{l}{J_{yy}} u_3(t) \\ \ddot{\psi} = \dfrac{\dot{\phi}\dot{\theta}\left(J_{yy} - J_{xx}\right)}{J_{zz}} + \dfrac{1}{J_{zz}} u_4(t) \end{cases} \tag{2}$$

$$\begin{cases} \ddot{x} = u_1(t) \dfrac{-\sin\theta\cos\phi\cos\psi - \sin\phi\sin\psi}{m} \\ \ddot{y} = u_1(t) \dfrac{\sin\phi\cos\psi - \sin\theta\cos\phi\sin\psi}{m} \\ \ddot{z} = -u_1(t) \dfrac{\cos\theta\cos\phi}{m} + g \end{cases} \tag{3}$$

where J_{RP} is the total moment of inertia of the motor rotor and propeller around the axis, and $\Omega_g = -\Omega_1 - \Omega_2 + \Omega_3 + \Omega_4$.

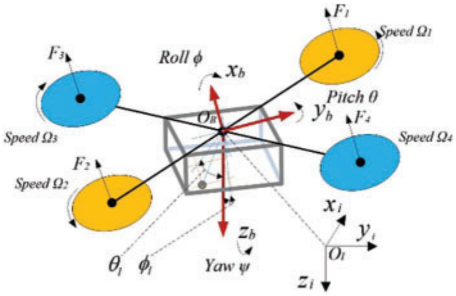

Figure 1: Quadrotor plant protection UAV

2.2 Problem Statement

In theory, a rigid multi-rotor aircraft can maintain its attitude stability by ensuring that each rotor provides equal thrust. Additionally, adjusting the thrust of the four rotors can change the position and attitude of the aircraft. Hence, based on the system dynamics model presented in Eqs. (2) and (3), the controller design can be carried out for the following affine nonlinear system:

$$\begin{cases} \dot{x}_1 (t) = x_2 (t) \\ \dot{x}_2 (t) = f (t) + g (t) \cdot u (t) \\ y(t) = x_1(t) \end{cases} \tag{4}$$

Among them, the state variables $x, \dot{x}, y, \dot{y}, z, \dot{z}, \phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}$ can be obtained through measurement and calculation, while$f (t)$ and $g (t)$ are the known system modeling parameters of the quadrotor vehicle system modeling process.

2.2.1 Model of Load Liquid Disturbance

Load disturbance can have a significant impact on UAV control during task execution. To mitigate this effect, various control methods have been proposed by researchers [8,34]. Similarly, the shaking of the aircraft water tank can cause interference with the stability of the aircraft's attitude. As shown in Fig. 2, the relative static model of the UAV on the x-axis plane at a certain moment during the movement of the liquid in the water tank can be observed.

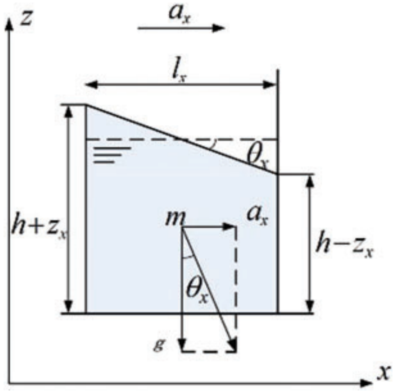

Figure 2: Static liquid model at the equilibrium moment

Fig. 2 illustrates parameters related to the systematic motion of liquid on the OXZ coordinate plane. Here, the UAV is assumed to move with a constant acceleration a_t (m/s^2), while the liquid system exhibits acceleration components along the x-axes and y-axes, denoted by a_x and a_y, respectively. The distances between the UAV's axis and the liquid's virtual centers of mass along the x-axes and y-axes are denoted by l_x and l_y, respectively. Here h is the distance (height) between the liquid level in the tank and the tank bottom at the initial time, g is gravity acceleration, θ_x and θ_y are the angles formed between the liquid level and the horizontal plane with respect to the x- and y-axes components, respectively. The parameters $h+z_x$ and $h-z_x$ are the heights of the contact surface between the liquid at equilibrium on the x-axis and the left and right container walls, respectively. On the y-axis, the corresponding heights are represented by $h + z_y$ and $h - z_y$, respectively.

At the initial moment, the forces of liquid formation in the x-axes and y-axes directions are derived as:

$$\begin{cases} P_x = \rho a_x l_x \\ P_y = \rho a_y l_y \end{cases} \tag{5}$$

where ρ is the liquid density, which remains constant.

Then, the forces exerted by the liquid along the x-axes and y-axes can be expressed as:

$$\begin{cases} F_x = P_x l_y h = \rho a_x l_x l_y h \\ F_y = P_y l_x h = \rho a_y l_x l_y h \end{cases} \tag{6}$$

At the moment of equilibrium, the force along the x and y-axes is the sum of the force exerted on the swaying part and the force exerted on the relatively stationary part: $\begin{cases} F_x = F_{x1} + F_{x2} \\ F_y = F_{y1} + F_{y2} \end{cases}$

At the moment of balance during the movement of the UAV, the force on the water tank along the x-axes and y-axes is the sum of the force on the water tank caused by the liquid sloshing and the force at the relatively static moment, denoted as $\begin{cases} F_x = F_{x1} + F_{x2} \\ F_y = F_{y1} + F_{y2} \end{cases}$. Consequently, the force on the relatively stationary part along the x-axes and y-axes can be mathematically represented as:

$$\begin{cases} F_{x2} = \rho l_x l_y \left(h - l_x \tan\theta_x\right) \ddot{x} \\ F_{y2} = \rho l_y l_x \left(h - l_y \tan\theta_y\right) \ddot{y} \end{cases} \tag{7}$$

Assuming that the acceleration of the liquid at the moment of equilibrium is equivalent to that of the drone, the force exerted by the shaking component can be expressed as follows when the liquid system in the tank reaches equilibrium:

$$\begin{cases} F_{x1} = \rho l_y \ddot{x} \int_0^{l_x \tan\theta_x} \dfrac{l_x \tan\theta_x - z_x}{l_x \tan\theta} d_z \\ F_{y1} = \rho l_x \ddot{y} \int_0^{l_y \tan\theta_y} \dfrac{l_y \tan\theta_y - z_y}{l_y \tan\theta_y} d_z \end{cases} \tag{8}$$

The maximum force components of the entire water tank along the x-axes and y-axes can be derived as follows:

$$\begin{cases} F_x = \rho l_x l_y \ddot{x} \left(h - \dfrac{3 l_x \tan\theta_x}{2}\right) \\ F_y = \rho l_y l_x \ddot{y} \left(h - \dfrac{3 l_y \tan\theta_y}{2}\right) \end{cases} \tag{9}$$

The impact of the liquid in the tank on the rigid structure of the aircraft can be quantified by the displacement of the liquid's center of mass within the container. This displacement is denoted as the movement distance $L = \begin{bmatrix} L_x & L_y \end{bmatrix}^T$, representing the perturbation force magnitude:

$$\begin{cases} L_x = \dfrac{\ddot{x} t^2}{2} - \dfrac{l_x}{2} + \dfrac{l_x^2}{12h} \tan\theta_x \\ L_y = \dfrac{\ddot{y} t^2}{2} - \dfrac{l_y}{2} + \dfrac{l_y^2}{12h} \tan\theta_y \end{cases} \tag{10}$$

Accordingly, the disturbance force magnitude that affects the aircraft in the pitch and roll attitude directions due to the liquid in the tank can be calculated as follows:

$$\begin{cases} F_\theta = \sqrt{2} F_x L_x l \\ F_\phi = \sqrt{2} F_y L_y l \end{cases} \tag{11}$$

The disturbance parameters for pitch and roll attitudes can be defined as follows:

$$d_{load} = \begin{bmatrix} d_\theta \\ d_\phi \end{bmatrix} = \begin{bmatrix} \dfrac{l}{J_{xx}} F_\theta \\ \dfrac{l}{J_{yy}} F_\phi \end{bmatrix} = \begin{bmatrix} \dfrac{\sqrt{2}}{J_{xx}} F_x L_x l^2 \\ \dfrac{\sqrt{2}}{J_{yy}} F_y L_y l^2 \end{bmatrix} \tag{12}$$

2.2.2 Actuator Fault Model

In general, in the absence of external disturbances, the lift of a multi-rotor aircraft is provided by the lift generated by each propeller, and the lift F_i produced by a single propeller rotation is linearly related to the square of the motor speed Ω_i^2. However, in practical operating environments, as the load increases or the attitude of the aircraft changes significantly, the performance of the motor may deteriorate with increasing temperature, and there may be a nonstrictly linear relationship between the motor control signal given by the controller and the motor speed. Considering the operation of the crop spraying unmanned aerial vehicle in farmland or orchard environments, the motor rotor is susceptible to contamination by dust or other debris, and the propeller blades may collide with plant roots and tree branches. In these cases, the multi-rotor UAV motor and propeller mechanical structure are prone to malfunctions (referred to collectively as actuator faults). Therefore, consider re-expressing the pulling force of a single motor as:

$$F_{f,i} = c_{T,i} \cdot \left(b_i^* (t) \cdot \kappa_{gain,i} (t) \cdot \Gamma_i + c_i (t) + b_i (t)\right)^2, (i = 1, \ldots, 4) \tag{13}$$

where $b_i^* (t)$ and $c_i (t)$ are the gain fault and the additive fault acting on a single motor, respectively. Due to the nonlinear relationship between lift $F_{f,i}$ and the speed squared Ω_i^2, the time function $\kappa_{gain,i} (t)$ is defined as the gain between the speed of a single motor and the motor controller signal Γ_i.

However, it is difficult to directly obtain the change data of the nonlinear term and actuator fault expressed in the mathematical model in the actual system. Therefore, to simplify the modeling process and facilitate the parameter estimation of the observer designed later. For the convenience of mathematical expression, the following text will ignore the expression form of the independent variable of time. According to Eqs. (1) and (4), the control quantity $u(t)$ with actuator fault is redefined as:

$$u_F = \alpha u + \tau \tag{14}$$

where $\alpha (t) \in (0, 1]$ is the parameter containing the actuator gain fault and uncertain nonlinearity, and τ is the parameter containing the actuator additive fault. According to the attitude dynamic model of the four-rotor aircraft in Eq. (2), which includes the effect of load disturbance, the nonlinear system model of the plant protection UAV can be simplified into a second-order control system.

Here, u_F represents the fault control law, $\alpha (t) \in (0, 1]$ is the parameter that includes the actuator gain fault and nonlinear uncertainty of the system, and T is the parameter that includes the actuator additive fault. Based on the attitude dynamic model of the quadrotor aircraft in Eq. (2), considering the load disturbance model added to the aircraft model, the nonlinear system model of the plant protection UAV can be simplified into a second-order control system.

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f_0 + \Delta f + (g_0 + \Delta g)\, u_F + d_{playload} \\ y = x_1 \end{cases} \tag{15}$$

In the above equation, f_0 is the deterministic term in the system that includes attitude angular velocities and modelling parameters, while g_0 is the modelling parameter term. Notably, f_0 and g_0 can be obtained through sensor measurements and experimental data, while Δf and Δg are parameter error terms in the model. By substituting Eq. (14) into the above equation and defining $\varsigma_1 = \Delta f + g_0\tau + \Delta g\tau$ as the parameter term containing actuator gain faults and model uncertainties and $\varsigma_2 = (g_0 + \Delta g)\,\alpha - g_0$ as the parameter term containing actuator bias faults and model uncertainties, an affine nonlinear system with wind disturbance, actuator faults, and model uncertainties can be obtained.

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f_0 + \Im_1 + g_0 u + \Im_2 u + d_{playload} \\ y = x_1 \end{cases} \tag{16}$$

3 Controller Design

3.1 Second-Order ADRC Design

The state space representation of a second-order system model of the UAV given in Eq. (16) can be expressed as follows:

$$\begin{cases} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}}_{A} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ g_0 \\ 0 \end{bmatrix}}_{B} \cdot u + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \cdot \dot{f} \\ y = x_1 = \underbrace{\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}}_{C} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \end{cases} \tag{17}$$

In Eq. (16), the fault and disturbance parameter terms in the system are quite complex. If an RBFNN is used to identify each parameter part, it will complicate the observer design and increase the computation of the controller. Therefore, for the convenience of the subsequent design of the ADRC observer, we further optimize and introduce a generalized disturbance term f. The second-order attitude model of the system in the state-space model can be expressed as:

$$\dot{x}_2 = f + g_0 u \tag{18}$$

Here, the generalized disturbance term f actually contains system state parameters, load disturbance $d_{playload}$, fault and the uncertain part of nonlinear system modelling with respect to time t, which can be expressed as:

$$f = f_0 + \varsigma_1 + d_{playload} + \varsigma_2 u \tag{19}$$

The basic idea in the design process of an active disturbance rejection controller (ADRC) is to develop an extended state observer (ESO) capable of estimating both the system state and total disturbance. The generalized disturbance term has been defined in Eq. (19). To estimate the generalized disturbance term f, it is considered as the extended state x_3. Thus, $\hat{x}_1 = \hat{y}$, $\hat{x}_2 = \dot{\hat{y}}$, $\hat{x}_3 = \hat{f}$, as shown in Fig. 3.

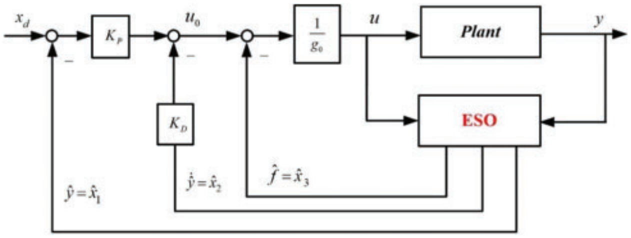

Figure 3: ADRC ring structure for second-order processes

The linear controller can achieve disturbance suppression and integral chain behavior by using estimation variables, and the extended state observer of the system can be obtained as follows:

$$\begin{bmatrix} \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \\ \dot{\hat{x}}_3 \end{bmatrix} = A \cdot \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \end{bmatrix} + B \cdot u + \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} \cdot \left(y - \hat{x}_1\right) \tag{20}$$

where $\begin{bmatrix} d_1 & d_2 & d_3 \end{bmatrix}^T$ can be defined as D, so the extended observer can be further expressed in the following form:

$$\begin{bmatrix} \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \\ \dot{\hat{x}}_3 \end{bmatrix} = (A - DC) \cdot \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \end{bmatrix} + B \cdot u + D \cdot y \tag{21}$$

As mentioned above, we have implemented a residual double integral behavior with disturbance suppression and a linear controller. Then, we can obtain a controller based on extended state feedback.

$$u = \frac{u_0 - \hat{x}_3}{g_0} \tag{22}$$

Therefore, according to the second-order ADRC structure designed in Fig. 3, $u_0(t)$ can be expressed as:

$$u_0 = K_P \cdot \left(x_d - \hat{y}\right) - K_D \cdot \dot{\hat{y}}, \tag{23}$$

where x_d is the reference input of the controlled system. Assuming that the estimated value of the ESO is relatively accurate, that is

$$\begin{cases} \hat{x}_1 = \hat{y} \approx y \\ \hat{x}_2 = \dot{\hat{y}} \approx \dot{y} \\ \hat{x}_3 = \hat{f} \approx f \end{cases} \tag{24}$$

Replacing the above equation in (18), the second-order system model can be expressed as $\dot{x}_2 = \left(f - \hat{x}_3\right) + u_0 \approx K_P \cdot (x_d - y) - K_D \cdot \dot{y}$. Furthermore, under ideal conditions, the system can be expressed as follows:

$$x_d = \frac{1}{K_P} \cdot \ddot{x}_1 + \frac{K_D}{K_P} \cdot \dot{x}_1 + x_1 \tag{25}$$

In general, the closed loop can be adjusted to the critically damped state with K_P and K_D for any second-order dynamics, and for the desired 2% adjustment time T_{settle}, a negative real bipolar can be obtained by selecting K_P and K_D, $s_{1/2}^{CL} = s^{CL}$.

Typically, for any second-order dynamic system, a PD controller can be used to adjust the closed loop to the critical damping state, and for the desired 2% settling time T, a negative real pole can be obtained by selecting proportional parameter K_P and derivative parameter K_D, defining $s_{1/2}^{CL} = s^{CL}$. Furthermore, it can be obtained that

$$\begin{cases} K_P = \left(s^{CL}\right)^2 \\ K_D = -2 \cdot s^{CL} \\ s^{CL} \approx -\dfrac{6}{T} \end{cases} \tag{26}$$

The placement of the poles of the ESO can be determined empirically and typically take the following values:

$$\begin{cases} s_{1/2/3}^{ESO} = s^{ESO} \approx (3 \ldots 10) \cdot s^{CL} \\ s^{CL} \approx -\dfrac{6}{T} \end{cases} \tag{27}$$

After the pole positions are thus selected, the observer gain is calculated according to the characteristic polynomial of the matrix $(A - DC)$. First, let us redefine the expression in terms of $\det(sI - (A - DC)) = s^3 + d_1 \cdot s^2 + d_2 \cdot s + d_3$, then

$$\begin{aligned} \det(sI - (A - DC)) &= \left(s - s^{ESO}\right)^3 \\ &= s^3 - 3s^{ESO} \cdot s^2 + 3s^{ESO^2} \cdot s - s^{ESO^3} \end{aligned} \tag{28}$$

Then, the solution for d_1, d_2, and d_3 can be factored out as follows:

$$\begin{cases} d_1 = -3s^{ESO} \\ d_2 = 3s^{ESO^2} \\ d_3 = -s^{ESO^3} \end{cases} \tag{29}$$

In summary, the control law of the ADRC controller for a class of nonlinear second-order systems can be designed as follows:

$$u = \frac{1}{g_0}\left(K_P \cdot \left(x_d - \hat{x}_1\right) - K_D \cdot \hat{x}_2 - \hat{x}_3\right) \tag{30}$$

3.2 Design of the ADRC Expanded State Observer Based on a Neural Network

The effectiveness of the attitude controller is crucial for the flight performance of plant protection unmanned aerial vehicles, especially in the presence of possible actuator faults. However, the control law proposed in this paper, which incorporates an estimate of the generalized disturbance term, can only achieve stable attitude control when the drone has a small degree of fault, as discussed in the experimental section. When the UAV (drone) has a large load, the proposed control law cannot achieve stable attitude control. Therefore, further research is needed to develop more robust control laws to handle larger disturbances and actuator faults. This article further improves on the previously designed fault-tolerant controller [31]. Therefore, this paper proposes a novel approach to estimate the generalized disturbance term in identifying a class of second-order nonlinear systems called the fractional gradient descent (FGD) learning algorithm. The proposed algorithm is illustrated in Fig. 4, which is incorporated into an ADRC controller structure.

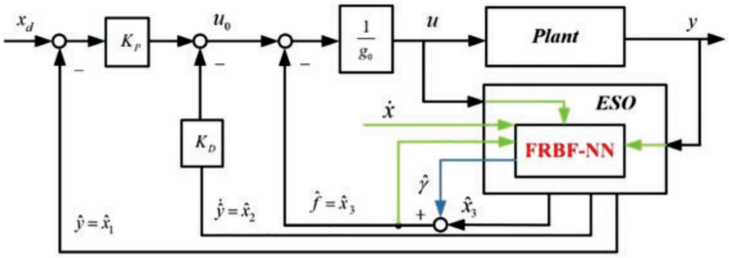

Figure 4: ADRC controller based on FRBF-NN

The experimental results demonstrate that the proposed method is effective in convergence and steady-state performance compared to the classical RBF neural network. The FRBF-NN proposed in this paper is a mathematical variant of the classical RBF-NN, and an algorithmic improvement is proposed in the intrinsic concept of gradient descent optimization.

The structural diagram of the proposed RBFNN is presented in Fig. 5.

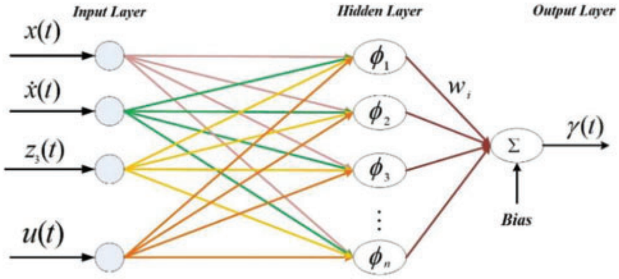

Figure 5: The FRBF-NN structural diagram

The RBF network structure consists of three layers: the input layer, the nonlinear hidden layer, and the linear output layer, as shown in Fig. 5. If we consider the input vector as $\mathrm{x} \in \mathbb{R}^4$, $\mathrm{x} = \begin{bmatrix} x & \dot{x} & z_3 & u \end{bmatrix}$, then the overall mapping of the RBF network $s : \mathbb{R}^{m_0} \rightarrow \mathbb{R}^1$ in the $n - th$ round learning iteration can be expressed as follows:

$$\gamma (n) = \sum_{i=1}^{m} \mathrm{w}_i (n) \phi_i (||\mathrm{x} - \mathrm{x}_i||) + b (n) \tag{31}$$

In the RBFNN structure shown in Fig. 5, m represents the number of hidden layer neurons, $x_i \in \mathbb{R}^{m4}$ represents the center of the RBFNN, $\mathrm{w}_i (n)$ represents the synaptic weight value of hidden layer neurons and output neurons, $b (n)$ represents the bias term of output neurons, and ϕ_i represents the basis function of the i-th hidden layer neuron. For simplicity, let us consider a single output neuron. Traditional RBF networks use multiple kernels, such as multiple quadric surfaces, inverse multiple quadric surfaces, and Gaussian kernel functions, which are considered to be the most commonly used kernel functions due to their versatility.

$$\phi_i (||\mathrm{x} - \mathrm{x}_i||) = \exp \left(\frac{- ||\mathrm{x} - \mathrm{x}_i||^2}{\sigma^2} \right), \tag{32}$$

where σ determines the diffusion value of the Gaussian kernel. In some cases, RBF uses the idea of measuring distances between network centers. Traditionally, Euclidean distance has been widely used as an effective distance measurement tool. However, by using the complementary properties between cosine distance measurement and Euclidean distance measurement, better results can be achieved [37].

$$\phi_i\left(\mathrm{x}.\mathrm{x}_i\right) = \frac{\mathrm{x}\cdot\mathrm{x}_i}{||\mathrm{x}||\,||\mathrm{x}_i|| + \kappa} \tag{33}$$

where $\kappa > 0$ is a very small constant, added to the denominator, to avoid taking an indeterminate form when $||\mathrm{x}||$ or $||\mathrm{x}_i||$ is 0. The cost function measures the difference between the actual output and the desired output for a given input and is used to evaluate the performance of the network during training:

$$\varepsilon\left(n\right) = \frac{1}{2}\left(d\left(n\right) - \gamma\left(n\right)\right)^2 = \frac{1}{2}\sum_k e_k^2\left(n\right) \tag{34}$$

where $d\left(n\right)$ is the desired output at the *n-th* iteration, $e\left(n\right) = d\left(n\right) - y\left(n\right)$ is the instantaneous error between the desired output and the actual output of the neuron, and k is the number of output neurons.

The weight updating equation of the gradient descent method with a mixed parameter q and a fractional gradient term is given as follows:

$$\mathrm{w}_i\left(n+1\right) = \mathrm{w}_i\left(n\right) - q\hbar\nabla_{\mathrm{w}_i}\varepsilon\left(n\right) - \left(1-q\right)\alpha\hbar_\upsilon\nabla^\upsilon_{\mathrm{w}_i}\varepsilon\left(n\right) \tag{35}$$

The parameters in this equation include $0 \le q \le 1$, $\hbar$, and $\hbar_\upsilon$, which represent the step sizes of the conventional gradient and the fractional gradient, respectively. Thus, the weight update rule using gradient descent can be expressed as follows:

$$\Delta\mathrm{w}_i\left(n\right) = -q\hbar\nabla_{\mathrm{w}_i}\varepsilon\left(n\right) - \left(1-q\right)q\hbar_\upsilon\nabla^\upsilon_{\mathrm{w}_i}\varepsilon\left(n\right) \tag{36}$$

We use the chain rule to compute the factors $-\nabla_{\mathrm{w}_i}\varepsilon\left(n\right)$, then

$$-\nabla_{\mathrm{w}_i}\varepsilon\left(n\right) = -\frac{\partial\varepsilon\left(n\right)}{\partial e_k\left(n\right)} \times \frac{\partial e_k\left(n\right)}{\partial y_k\left(n\right)} \times \frac{\partial y_k\left(n\right)}{\partial\mathrm{w}_i\left(n\right)} \tag{37}$$

Similarly, the fractional derivative of ladder $\nabla^\upsilon_{\mathrm{w}_i}\xi\left(n\right)$ can be calculated by applying the chain rule of fractional calculus, i.e., $D^\upsilon_x\upsilon\left(g\left(x\right)\right) = \left(D^1_x\upsilon\left(g\right)\right)_{g=g(x)} D^\upsilon_x\upsilon\left(g\right)$.

$$-\nabla^\upsilon_{\mathrm{w}_i}\xi\left(n\right) = -\frac{\partial\varepsilon\left(n\right)}{\partial e_k\left(n\right)} \times \frac{\partial e_k\left(n\right)}{\partial y_k\left(n\right)} \times \frac{\partial^\upsilon_k y_k\left(n\right)}{\partial\mathrm{w}^\upsilon_i\left(n\right)} \tag{38}$$

By simplifying the two formulas above, we obtain

$$\begin{cases} -\nabla_{\mathrm{w}_i}\varepsilon\left(n\right) = \phi_i\left(\mathrm{x},\mathrm{x}_i\right)e_k\left(n\right) \\ -\nabla^\upsilon_{\mathrm{w}_i}\xi\left(n\right) = \phi_i\left(\mathrm{x},\mathrm{x}_i\right)e_k\left(n\right)\dfrac{w_i^{1-\upsilon}\left(n\right)}{\Gamma\left(2-\upsilon\right)} \end{cases} \tag{39}$$

Then, formula $\mathrm{w}_i\left(n+1\right) = \mathrm{w}_i\left(n\right) - \alpha\hbar\nabla_{\mathrm{w}_i}\varepsilon\left(n\right) - \left(1-\alpha\right)\alpha\hbar_\upsilon\nabla^\upsilon_{\mathrm{w}_i}\varepsilon\left(n\right)$ can be reduced to

$$\mathrm{w}_i\left(n+1\right) = \mathrm{w}_i\left(n\right) + q\hbar\phi_i\left(\mathrm{x},\mathrm{x}_i\right)e_k\left(n\right) + \left(1-q\right)\hbar_\upsilon\phi_i\left(\mathrm{x},\mathrm{x}_i\right)e_k\left(n\right)\frac{w_i^{1-\upsilon}\left(n\right)}{\Gamma\left(2-\upsilon\right)} \tag{40}$$

To simplify the computation, we consider $\mu_\upsilon \equiv \mu_\upsilon\Gamma\left(2-\upsilon\right)$, which yields

$$\begin{aligned} \mathrm{w}_i\left(n+1\right) &= \mathrm{w}_i\left(n\right) + q\hbar\phi_i\left(\mathrm{x},\mathrm{x}_i\right)e_k\left(n\right) + \left(1-q\right)\hbar_\upsilon\phi_i\left(\mathrm{x},\mathrm{x}_i\right)e_k\left(n\right)w_i^{1-\upsilon}\left(n\right) \\ &= \mathrm{w}_i\left(n\right) + e_k\left(n\right)\left(q\hbar + \left(1-q\right)\hbar_\upsilon w_i^{1-\upsilon}\left(n\right)\right)\phi_i\left(\mathrm{x},\mathrm{x}_i\right) \end{aligned} \tag{41}$$

The update rule for $b\left(n\right)$ can be expressed as follows:

$$b\left(n+1\right)=b\left(n\right)+e_k\left(n\right)\left(q\hbar+\left(1-q\right)\hbar_\upsilon b^{1-\upsilon}\left(n\right)\right) \tag{42}$$

4 Design of Fault-Tolerant Controller and Proof of Stability

In a multi-rotor control system, the control output for the pitch and roll attitude of the aircraft exceeds its input, making it a typical underactuated system. In the event of a failure, the stability of the attitude system becomes paramount. Therefore, this section primarily focuses on designing the controller for the aircraft's attitude system.

First, based on the affine nonlinear system given in Eq. (16), we define x_d as the angle reference signal of the attitude controller, which is a continuously differentiable time series signal. Additionally, the tracking error of the system is defined as follows:

$$\begin{cases} e_1=x_1-x_d \\ e_2=K_pe_1-K_Dx_2 \\ e_3=e_2-f \end{cases} \tag{43}$$

where e_1 and e_2 are the tracking errors of the system attitude angle and angular velocity, respectively. For ease of notation, some parameter arguments will be omitted in the subsequent derivations and replaced with parameter symbols.

To achieve stable control over x_1, the Lyapunov function V_1 is defined as:

$$V_1=\frac{1}{2}e_1^2 \tag{44}$$

The derivative of time with respect to that is:

$$\begin{aligned} \dot{V}_1&=e_1\dot{e}_1 \\ &=e_1\left(x_2-\dot{x}_d\right) \\ &=e_1\left(\mu_1-e_2-\dot{x}_d\right) \\ &=-k_1e_1^2+e_1e_2 \end{aligned} \tag{45}$$

From Eq. (43), $\dot{V}$ can also be expressed as:

$$\dot{V}_1=e_1\left(x_2-\dot{x}_d\right) \tag{46}$$

Therefore, the error variable can be defined by introducing the virtual control quantity μ_1, defined as the error variable

$$e_2=\mu_1-x_2 \tag{47}$$

Then, the Eq. (46) can be expressed as:

$$\dot{V}_1=e_1\left(\mu_1-e_2-\dot{x}_d\right) \tag{48}$$

Therefore, when the virtual control quantity μ_1 is $\mu_1=\dot{x}_d-k_1e_1$, $k_1>0$, we can obtain

$$\begin{aligned} \dot{V}_1&=e_1\left(e_2-k_1e_1\right) \\ &=-k_1e_1^2+e_1e_2 \end{aligned} \tag{49}$$

Then, based on the expression of Eq. (45), the time derivative of e_1 can be represented as:

$$\dot{e}_1 = -k_1 e_1 + e_2 \tag{50}$$

Then, according to Eq. (43), it is further derived

$$\begin{aligned} e_2 &= \dot{e}_1 + k_1 e_1 \\ &= \dot{x}_1(t) - \dot{x}_d(t) + k_1 e_1 \end{aligned} \tag{51}$$

However, to satisfy the Lyapunov function's stability condition $\dot{V}_1 < 0$, the unstable term $e_1 e_2$ in Eq. (45) needs to be eliminated. Therefore, we selected the new Lyapunov function $V_2 = V_1 + \frac{1}{2}e_2^2$ and obtain its derivative as follows:

$$\begin{aligned} \dot{V}_2 &= \dot{V}_1 + e_2\dot{e}_2 \\ &= -k_1e_1^2 + e_1e_2 + e_2\dot{e}_2 \\ &= -k_1e_1^2 + e_2(e_1 + \dot{e}_2) \end{aligned} \tag{52}$$

Then, $\dot{e}_2 = \dot{x}_2(t) - \ddot{x}_d(t) + k_1\dot{e}_1$ can be obtained from Eq. (52), which can be further derived after substituting into Eq. (53):

$$\begin{aligned} \dot{V}_2 &= -k_1e_1^2 + e_2(e_1 + \dot{x}_2(t) - \ddot{x}_d(t) + k_1\dot{e}_1) \\ &= -k_1e_1^2 + e_2(e_1 + f + g_0u - \ddot{x}_d + k_1\dot{e}_1) \\ &= -k_1e_1^2 + e_2(e_1 + e_2 - e_3 + g_0u - \ddot{x}_d + k_1\dot{e}_1) \\ &= -k_1e_1^2 + e_2(e_1 + e_2 + g_0u - \ddot{x}_d + k_1\dot{e}_1) - e_2e_3 \end{aligned} \tag{53}$$

Therefore, if $(-e_2e_3)$ is a stable term, take $e_1 + e_2 + g_0u - \ddot{x}_d + k_1\dot{e}_1 = -k_2e_2$, $k_2 > 0$, and we obtain the control law u that makes the system asymptotically stable. Therefore, we reselect the new Lyapunov function V_3 to be

$$V_3 = V_1 + \frac{1}{2}e_2^2 + \frac{1}{2}e_3^2 \tag{54}$$

Taking the derivative of the latter yields

$$\begin{aligned} \dot{V}_3 &= \dot{V}_2 + e_3\dot{e}_3 \\ &= -k_1e_1^2 - k_2e_2^2 - e_2e_3 + e_3\dot{e}_3 \\ &= -k_1e_1^2 - k_2e_2^2 + e_3(\dot{e}_3 - e_2) \\ &= -k_1e_1^2 - k_2e_2^2 + e_3(-k_2\dot{e}_2 - \ddot{e}_2 - \dot{e}_1 - e_2) \end{aligned} \tag{55}$$

To satisfy the Lyapunov function's stability condition $\dot{V}_3 < 0$, the following equalities need to hold:

$$\begin{cases} -k_2e_2 = e_1 + \dot{x}_2(t) - \ddot{x}_d(t) + k_1\dot{e}_1 \\ -k_3e_3 = -k_2\dot{e}_2 - \ddot{e}_2 - \dot{e}_1 - e_2 \end{cases} \tag{56}$$

Among them, $k_3 > 0$. Further, we get

$$\begin{aligned}-k_3 \left(e_2 \left(t\right) - f \left(t\right)\right) &= -k_2\dot{e}_2 - \ddot{e}_2 - \dot{e}_1 - e_2 \\ &= -k_2 \left(\dot{x}_2 \left(t\right) - \ddot{x}_d \left(t\right) + k_1\dot{e}_1\right) - \ddot{e}_2 - \dot{e}_1 - e_2 \\ &= -k_2\dot{x}_2 \left(t\right) + k_2\ddot{x}_d \left(t\right) - k_2k_1\dot{e}_1 - \ddot{e}_2 - \dot{e}_1 - e_2 \\ &= -k_2 \left(g_0u \left(t\right) + f \left(t\right)\right) + k_2\ddot{x}_d \left(t\right) - k_2k_1\dot{e}_1 - \ddot{e}_2 - \dot{e}_1 - \left(K_pe_1 \left(t\right) - K_Dx_2 \left(t\right)\right) \\ &= -k_2g_0u - k_2f + k_2\ddot{x}_d - k_2k_1\dot{e}_1 - \ddot{e}_2 - \dot{e}_1 - K_pe_1 + K_Dx_2 \end{aligned} \quad (57)$$

Finally, the control that makes the system asymptotically stable can be inversely solved

$$\begin{aligned}u &= \frac{1}{k_2g_0}\left(k_3e_2 - k_3f - k_2f + k_2\ddot{x}_d - k_2k_1\dot{e}_1 - \ddot{e}_2 - \dot{e}_1 - K_pe_1 + K_Dx_2\right) \\ &= \frac{1}{k_2g_0}\left(k_3e_2 - \left(k_3 + k_2\right)f - k_2k_1\dot{e}_1 + k_1e_1 - 2e_2 + k_2\ddot{x}_d - \ddot{e}_2\right) \\ &= \frac{1}{g_0}\left(\left(\frac{k_1}{k_2} - k_1^2\right)e_1 + \left(\frac{k_3}{k_2} - \frac{1}{k_2} - k_1\right)e_2 - \left(\frac{k_3}{k_2} + 1\right)f + \ddot{x}_d - \frac{1}{k_2}\ddot{e}_2\right)\end{aligned} \quad (58)$$

A multirotor UAV itself is a high-order system, and the introduction of higher-order variables can increase system instability. Therefore, $\left(\ddot{x}_d - \frac{1}{k_2}\ddot{e}_2\right)$ in the second-order variable of the system is omitted to achieve better control. Meanwhile, to reduce the potential jitter problem that may arise after omitting the higher-order terms, we let $\mathrm{O}_{high} = \ddot{e}_2 - k_2\ddot{x}_d$, which satisfies the following condition:

$$\mathrm{O}_{high} = \begin{cases}\ddot{e}_2 - k_2\ddot{x}_d & \ddot{e}_2 - k_2\ddot{x}_d > 0 \\ 1 & \ddot{e}_2 - k_2\ddot{x}_d \leq 0\end{cases} \quad (59)$$

We can obtain the fault-tolerant control law in which the higher-order variables are omitted:

$$u = \frac{1}{g_0}\left(\left(\frac{k_1}{k_2} - k_1^2\right)e_1 + \left(\frac{k_3}{k_2} - \frac{1}{k_2} - k_1\right)e_2 - \left(\frac{k_3}{k_2} + 1\right)f\right) \quad (60)$$

Furthermore, the final form of the Lyapunov function can be determined based on Eqs. (45) and (54):

$$V = \frac{1}{2}e_1^2 + \frac{1}{2}e_2^2 + \frac{1}{2}e_3^2 \quad (61)$$

Next, we take the derivative of the above function:

$$\begin{aligned}\dot{V} &= e_1\dot{e}_1 + e_2\dot{e}_2 + e_3\dot{e}_3 \\ &= -k_1e_1^2 - k_2e_2^2 - k_3e_3^2\end{aligned} \quad (62)$$

It is easy to see that if $k_1 > 0$, $k_2 > 0$ and $k_3 > 0$ and the fault-tolerant control law is the result from Eq. (58), the derivative of the Lyapunov function $\dot{V} < 0$. Now, the system's status tracking error gradually approaches zero, and the entire system tends to stabilize.

$$
\begin{aligned}
\dot{V} &= -k_1e_1^2 - k_2e_2^2 + \left(-k_2g_0u - k_2f + k_2\ddot{x}_d - k_2k_1\dot{e}_1 - \ddot{e}_2 - \dot{e}_1 - K_pe_1 + K_Dx_2\right)e_3 \\
&= -k_1e_1^2 - k_2e_2^2 + \left(-\left(k_3e_2 - k_3f\right) + \left(k_2\ddot{x}_d - \ddot{e}_2\right)\right)e_3 \\
&= -k_1e_1^2 - k_2e_2^2 - k_3e_3^2 + \left(k_2\ddot{x}_d - \ddot{e}_2\right)e_3 \\
&= -k_1e_1^2 - k_2e_2^2 - k_3e_3^2 - \mathrm{O}_{high}e_3
\end{aligned}
\tag{63}
$$

Similarly, the derivative of the Lyapunov function $\left(\ddot{x}_d - \frac{1}{k_2}\ddot{e}_2\right)$ still holds true after we omit the higher-order term in the fault-tolerant control law. In other words, the system is still stable after reducing the computational complexity of the fault-tolerant control law.

5 Simulation Results and Analysis

This section aims to compare the effectiveness of the proposed fault-tolerant control algorithm with traditional methods through flight experiments and simulation experiments, specifically on the Qball-X4 unmanned aerial vehicle (UAV) model.

Real-time control and monitoring during the experiment were accomplished through the TCP/IP communication protocol between the onboard computer and the computer-based ground station. The numerical values of the initial model parameters for the modified crop-spraying drone are shown in Table 1.

Table 1: Plant protection UAV parameters

Parameter value	Interpretation
$m = 26.8\,kg$	Aircraft mass
$g = 9.8\,m/s^2$	Acceleration of gravity
$J_{xx} = 2.280\,kg{\cdot}m^2$	
$J_{yy} = 2.280\,kg{\cdot}m^2$	Moment of inertia matrix $J = diag\left(J_{xx}, J_{yy}, J_{zz}\right)$
$J_{zz} = 3.857\,kg{\cdot}m^2$	
$d = 0.74\,m$	Multi-rotor fuselage radius (1/2 wheelbase)
$C_m = 0.00004162\,N{\cdot}m/(rad/s)^2$	Integrated moment coefficient of a single propeller, moment (N·m) divided by rotational speed (rad/s), i.e., ($C_m = M_P/\omega^2$)
$C_R = 291.33\,rad/s$	Motor curve: input throttle value σ $(0 \sim 1)$ to steady motor speed $\omega_{ss} = C_R * \sigma + \omega_b$
$C_b = 68.74\,rad/s$	
$J_m = 0.0212\,kg{\cdot}m^2$	Moment of inertia of motor propeller
$T_m = 0.02\,s$	Motor response time constant
$ky = 4\,N{\cdot}m$	
$K = 120\,N$	

As shown in Fig. 6, the fault injection system of the Qball-X4 UAV model used PIXHAWK4 as the UAV flight control unit, and the Jetson TX2 onboard computer served as the data acquisition and calculation unit for the neural network estimator.

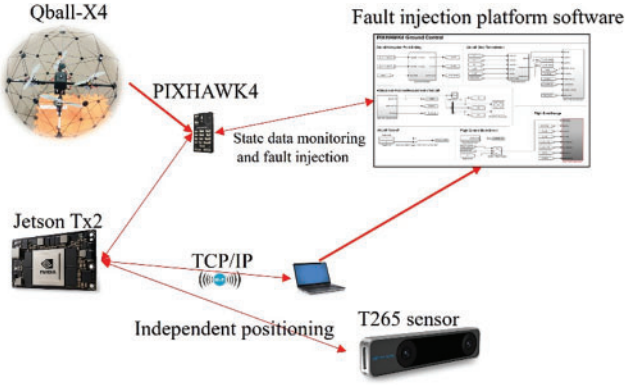
Qball-X4
Fault injection platform software
PIXHAWK4
State data monitoring and fault injection
Jetson Tx2
TCP/IP
Independent positioning
T265 sensor

Figure 6: Qball-X4 quadrotor UAV fault injection system

To effectively analyze the control performance of the proposed ADRC (RBFNN) fault-tolerant flight control algorithm, we collected system state data of the Qball-X4 UAV during actual flight, including the state matrix $[x, y, z, \phi, \theta, \psi]$, $\left[\dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi}\right]$, and control input matrix U. As the pitch and roll subsystems are the most common areas for controller design, this study mainly focuses on demonstrating the control performance of the Qball-X4 UAV in these subsystems during the experimental phase.

First, in the training process of the proposed gradient descent-based radial basis function neural network (FRBFNN) in this paper, to improve the fitting and accuracy of FRBFNN output parameter estimation with real data, flight state data of the aircraft were collected under normal, payload, and fault conditions.

We used the attitude state parameter $\left[\phi\ \psi\ \theta\ \dot{\phi}\ \dot{\theta}\ \dot{\psi}\right]$ and control input $[u_1\ u_2\ u_3\ u_4]$ collected during the flight of the Qball-X4 drone as the input for training the FRBFNN model. A raw state dataset was collected during the flight experiment, consisting of a combination of standard flight data and injected fault data. The drone's body sensors were set to sample the experimental data at a frequency of 1 ms, with each set of data containing 19405 continuous time series data and collected once every 90 s. The data were divided into training and testing sets in a 38:1 ratio. Two parallel neural network models, each with ten neurons, were designed, and a learning step size of 0.01 was set for gradient descent. The FRBFNN was iteratively trained using the weight updating rules designed in Section 3 of this paper. The trained FRBFNN network model was embedded into the Jetson TX2 onboard computer and directly communicated with PIXHAWK4 to properly track the expected signal. Fig. 7 shows that, based on a sample dataset containing fault data, the average mean square error obtained using the proposed FRBF-NN was superior to that of the traditional RBFNN, as validated through comparison. The control performance of the UAV was mainly demonstrated in the pitch and roll attitude subsystems during the experimental phase, as they are the most typical objects for controller design.

Figs. 8 and 9 show that the FRBF-NN has a more accurate tracking and prediction effect on the sample data than the traditional RBFNN, with slightly smaller tracking errors and error values tending toward zero.

In summary, we have developed an offline FRBF-NN model for plant protection UAVs with load disturbances and actuator faults, which provides a faster and more accurate estimation of fault and disturbance parameters than traditional RBFNN models. To further verify and compare the effectiveness and advantages of the proposed fault-tolerant control algorithm, we designed the following simulation scenarios to evaluate the controller's performance:

Scenario 1: An agricultural drone with a load disturbance of 8 kg but no actuator faults runs on the expected navigation path.

Scenario 2: An agricultural drone with actuator gain and bias faults but no load runs on the expected navigation path.

Scenario 3: An agricultural drone with actuator gain and bias faults and a load of 8 kg runs on the expected navigation path.

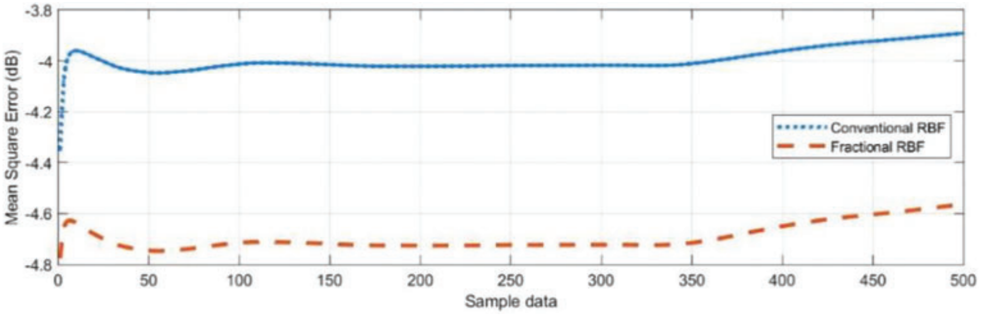

Figure 7: MSE testing curves

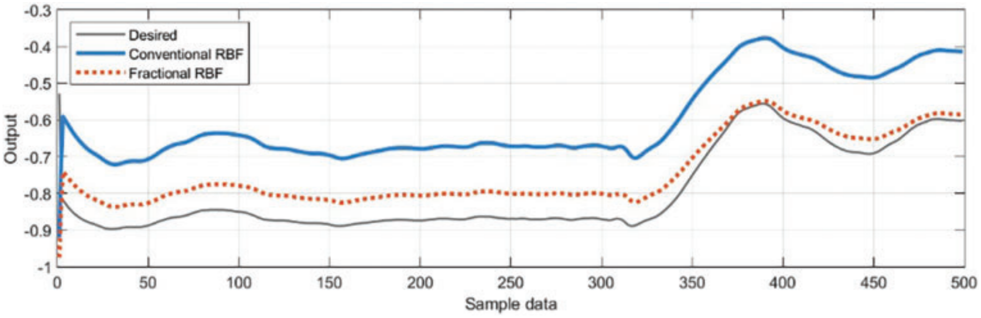

Figure 8: Predicted results of the FRBF-NN and traditional RBFNN

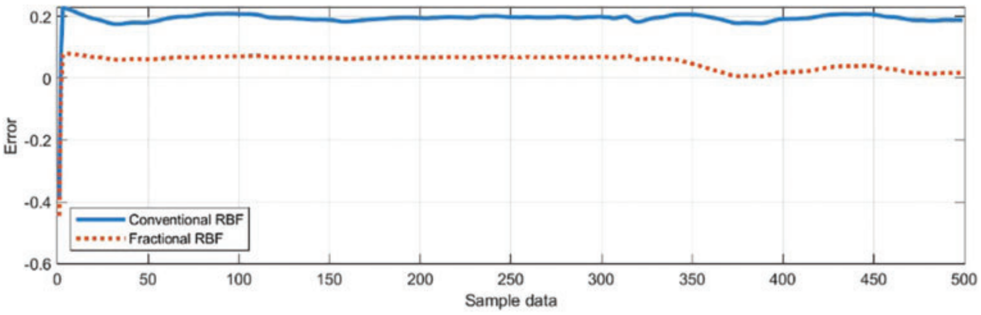

Figure 9: The prediction errors of the FRBF-NN and traditional RBFNN

First, as shown in Fig. 10, in scenario 1, the aircraft moves along the expected navigation path coordinates [0 0 −2; 4 0 −2; 4 4 −2; 0 4 −2; 0 0 −2; 0 0 0 0], and although the aircraft has body swing during the actual flight process, both the ADRC controller with expansion observer and the fault-tolerant controller proposed in this paper have good position tracking performance. From Figs. 11 and 12, it can be seen that under the payload condition, the tracking errors of the second-order ADRC controller, NNESO ADRC controller, and FRBFNN ADRC controller are all within the allowable control range. However, as seen from the attitude tracking and error curves of the unmanned aerial vehicle in Figs. 11 and 12, the tracking error of the flight controller with FRBF-NN is smaller, and the body swing amplitude is smaller.

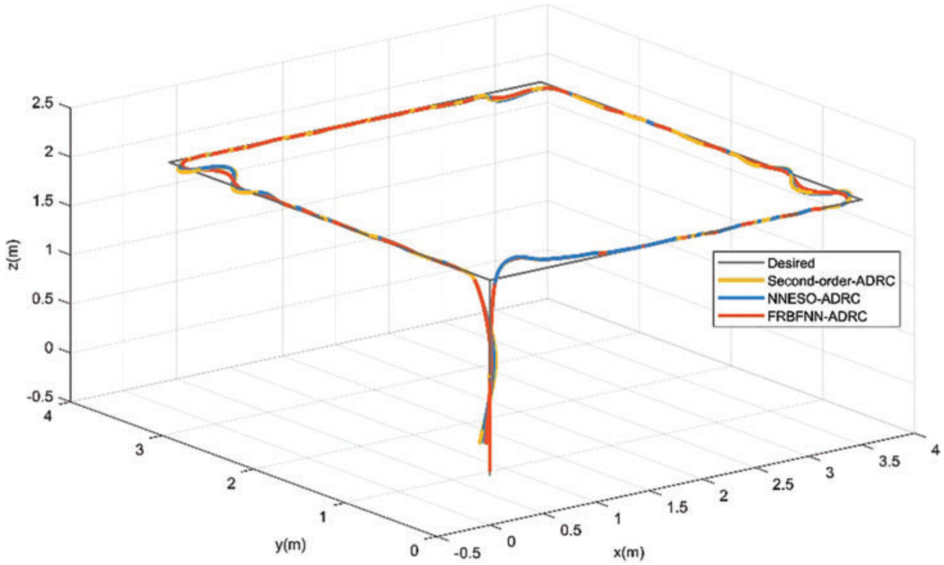

Figure 10: Position tracking of the UAV with a load of 8 kg and no faults

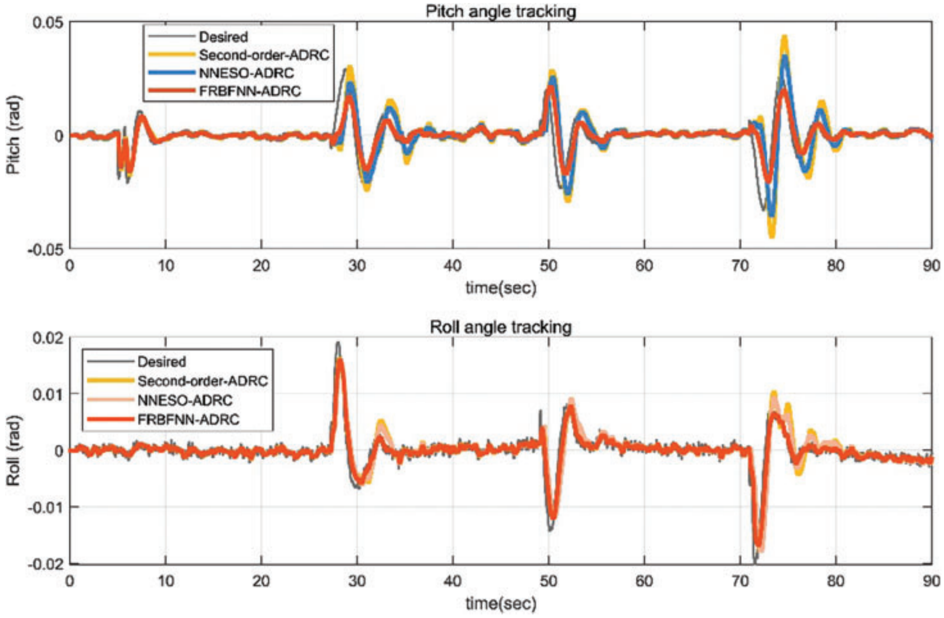

Figure 11: Pitch and roll angles of the UAV with an 8 kg load and no faults

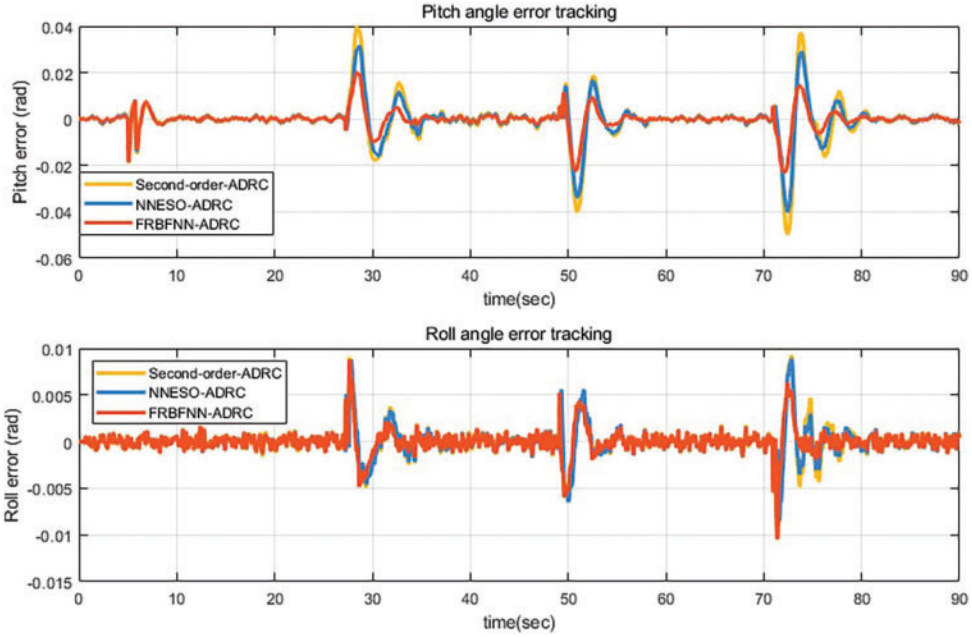

Figure 12: Pitch and roll angle errors of the drone with an 8 kg payload and no faults

In scenario 2, when the UAV unloaded the payload, we injected a 20% gain fault into motors 2 and 3 at the 35th second of flight and injected a 10% bias fault into motors 2 and 3 again at the 60th second. As shown in Fig. 13, both the second-order ADRC controller, NNESO ADRC controller and the proposed fault-tolerant controller still exhibit excellent path-tracking performance during flight. Figs. 14 and 15 reveal that the proposed fault-tolerant controller has smaller errors and better performance during the occurrence of faults.

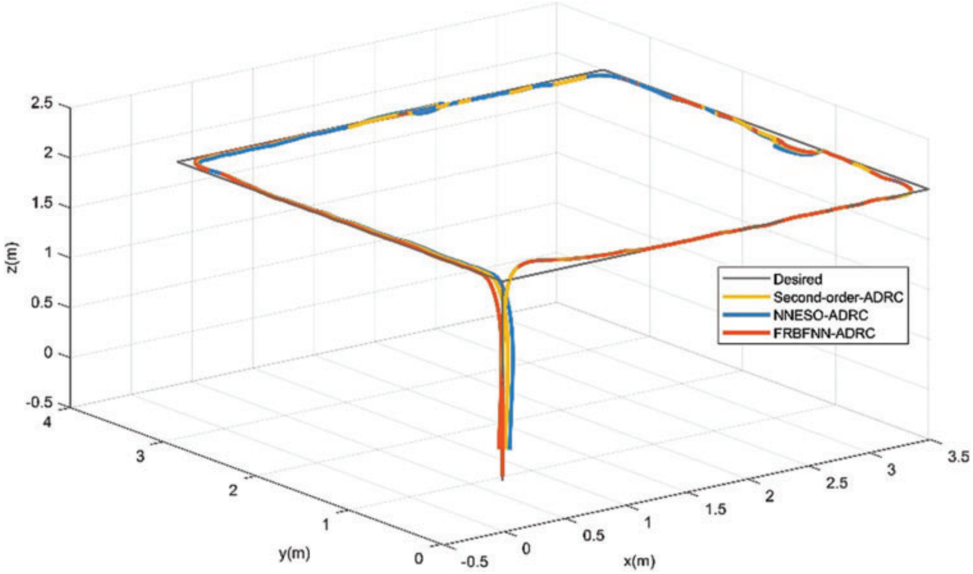

Figure 13: Position tracking of the UAV under actuator faults

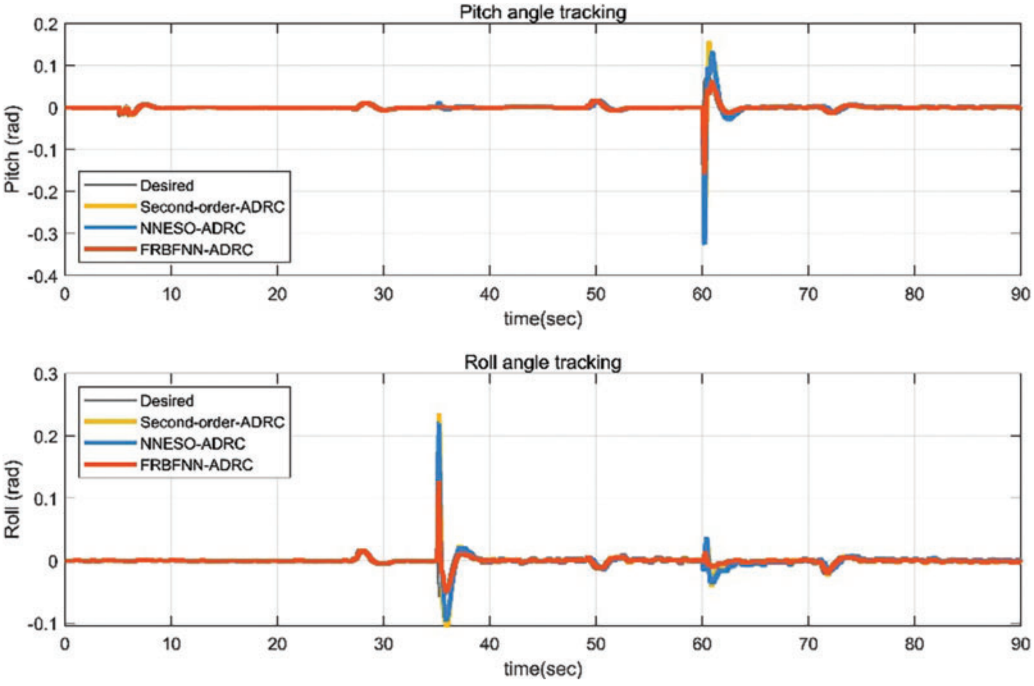

Figure 14: Pitch and roll angles of the UAV with actuator faults

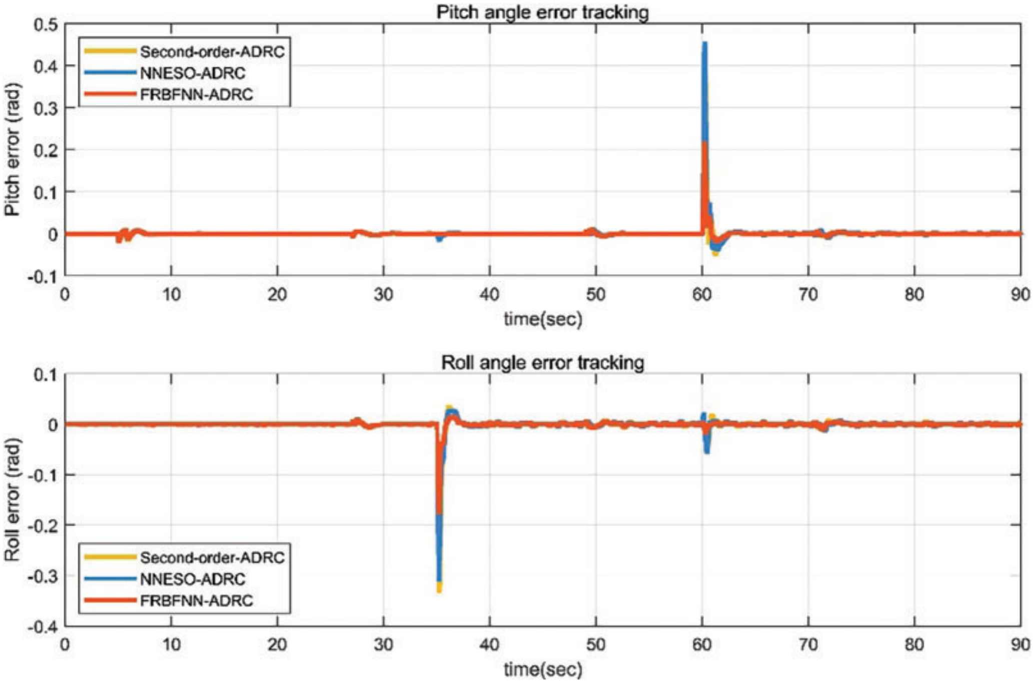

Figure 15: Pitch and roll angle errors of the UAV with actuator faults

In scenario 3, we simulated the situations of load and actuator failures in scenarios 1 and 2, as shown in Fig. 16. At this time, we can compare that the proposed fault-tolerant control algorithm has better path-tracking control and less aircraft oscillation than the second-order ADRC and NNESO ADRC. In Figs. 17 and 18, the attitude and angle tracking errors of the aircraft further demonstrate the more stable attitude control ability of the proposed fault-tolerant control algorithm. It can be seen that NNESO ADRC has a certain degree of fault tolerance. Surprisingly, due to the dual effects of payload disturbance and actuator failure, the controller attitude collapsed when the NNESO ADRC controller injected a bias fault.

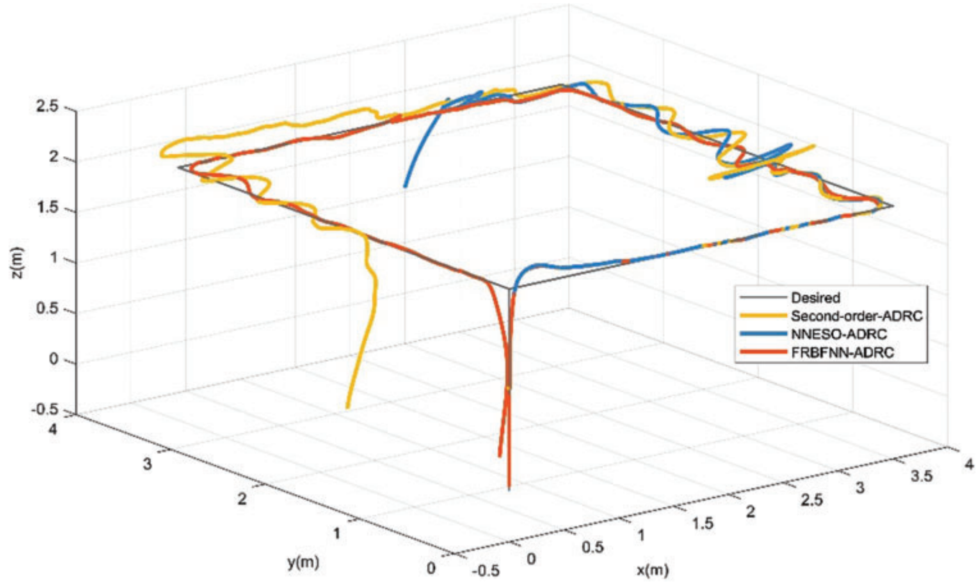

Figure 16: Position tracking of UAV with actuator faults and payload

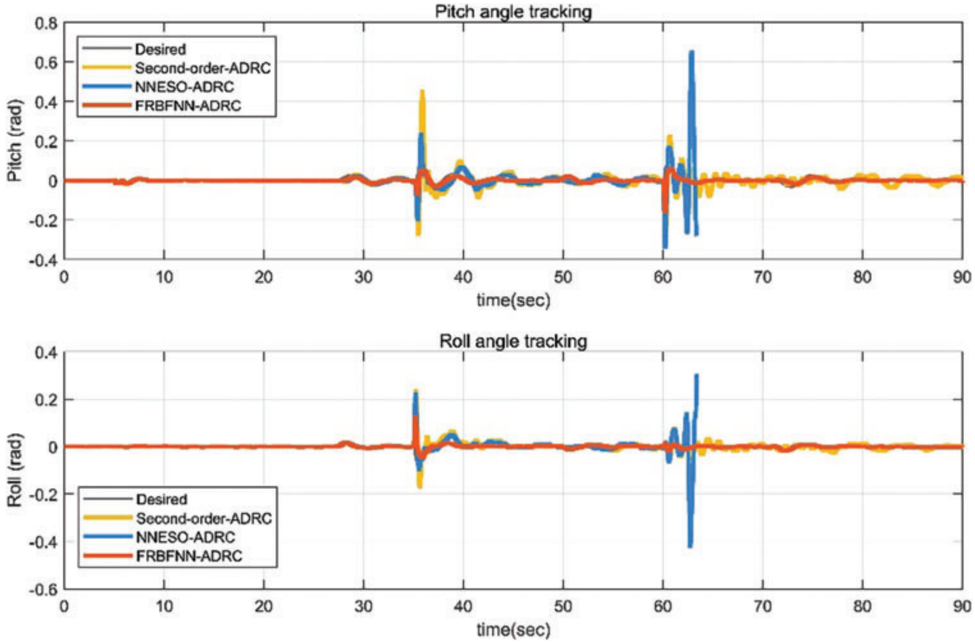

Figure 17: Pitch and roll angles of UAV with actuator faults and payload

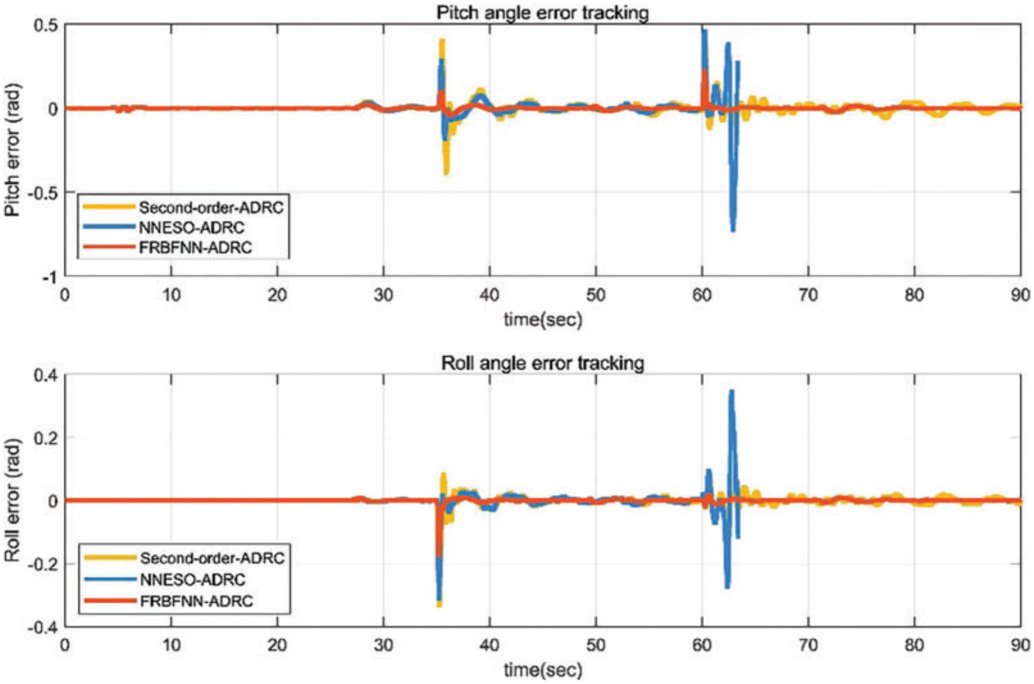

Figure 18: Pitch and roll angle errors of UAV with actuator faults and payload

The observer estimation curves of the pitch and roll subsystems shown in Figs. 19 and 20 are compared with the estimated value $\begin{bmatrix} z_1 & z_2 & z_3 \end{bmatrix}^T$ of the observer with the expanded state in the three scenarios listed based on the FRBF-NN-based fault-tolerant controller. They indicate that the proposed ESO based on FRBF-NN provides accurate estimation in time once the flight attitude experiences significant swing or when actuator faults occur. As shown in Fig. 21, it can be concluded that combining the onboard computer with higher computing power with the flight control system yields an excellent control effect.

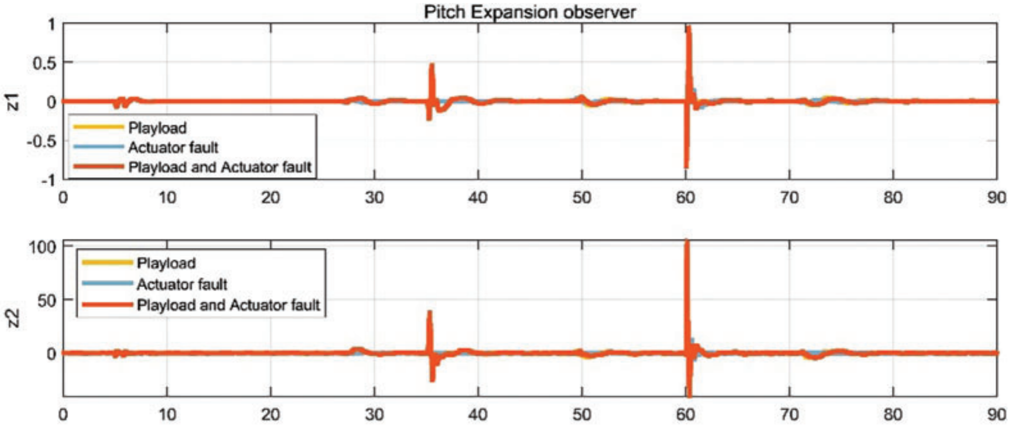

Figure 19: (Continued)

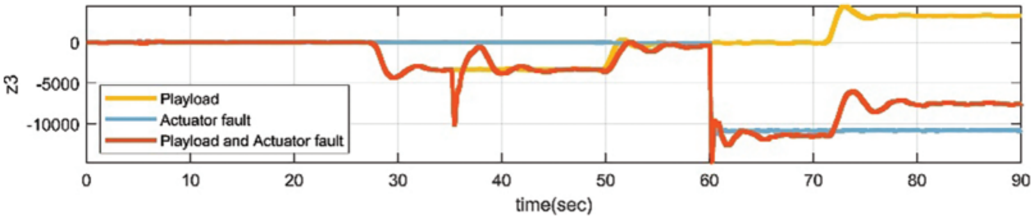

Figure 19: Estimated values of the pitch axis observer in three scenarios for the UAV

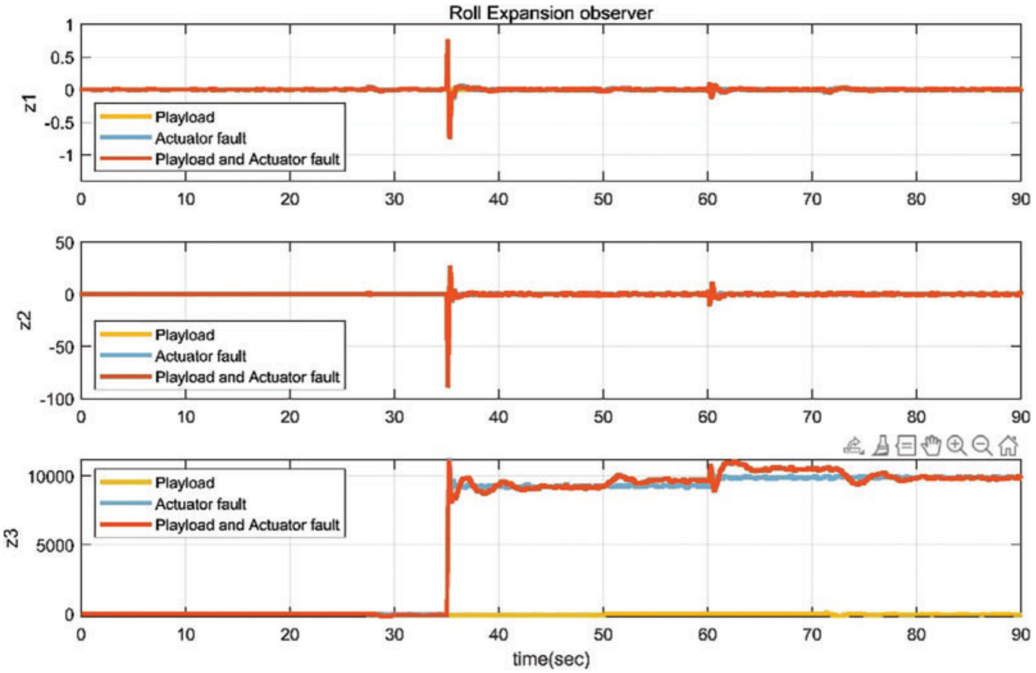

Figure 20: Estimated values of roll axis observer in three scenarios for the UAV

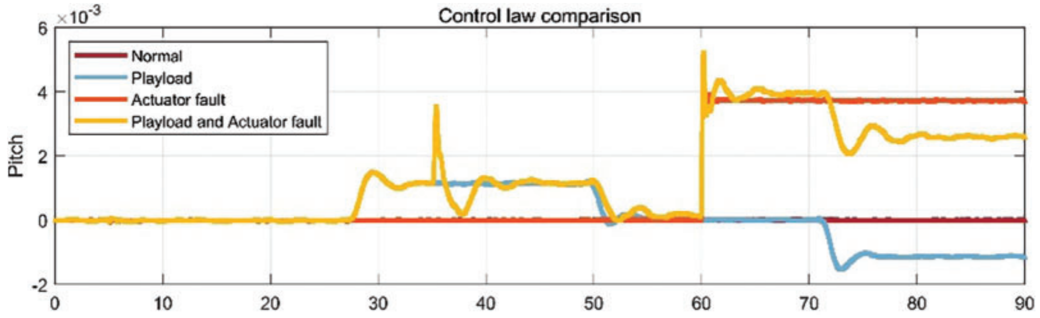

Figure 21: (Continued)

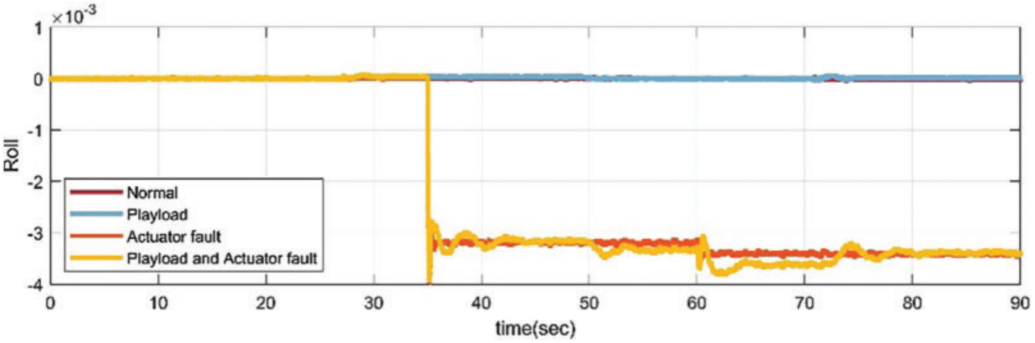

Figure 21: Comparison of pitch and roll control laws for the UAV

Finally, comprehensively analyze the above three scenarios. Comparing the second-order ADRC controller based on traditional RBFNN with the NNESO ADRC controller, it is not difficult to find that both have excellent performance in anti-interference ability. Moreover, in the event of a malfunction, the response speed of each controller is equivalent. However, in terms of accuracy comparison of errors, the FRBFNN ADRC proposed in this paper has smaller errors.

6 Conclusions

This article presents a novel active fault-tolerant control method based on FRBFNN and a second-order ADRC observer to address actuator bias faults, gain faults, and load disturbance issues in UAVs. The simulation results demonstrate that the designed ADRC controller with playload disturbance suppression capability is a powerful control tool, outperforming other controllers and integrating the uncertain parameters of the actuator model and load disturbance parameters. Furthermore, a new design framework of a self-disturbance fault-tolerant control method with FRBFNN is proposed by combining the traditional and fractional gradient descent methods, taking advantage of the complementary properties of the two. The algorithm's validity is verified on a multirotor fault-tolerant experimental platform and shows high convergence performance. Comparing the experimental results, it can be found that the proposed fault-tolerant control method has a response speed that is not weaker than other controllers, and has more accurate control tracking ability.

However, the proposed method has certain limitations. Its ability to fully exploit its potential in specific applications depends on the relationship between process dynamics, observer dynamics, sampling time, and measurement noise. On the one hand, the observer must be sufficiently fast compared to the process and closed-loop dynamics to provide more accurate observation values. For example, using an onboard computer for calculating the output of the FRBFNN in this article reduces the impact of the insufficient computing capacity of PIXHAWK4 and greatly improves the accuracy and speed of observer output data. On the other hand, the value of expanding observer poles will be limited by the effect of system sampling frequency and noise on control, which requires a good compromise. In addition, the controller proposed in this article has varying degrees of fault tolerance when actuator failures occur in aircraft with different physical parameter models. Therefore, the follow-up study will seek alternative optimized replacement methods.

Acknowledgement: None.

Funding Statement: This research was funded by the 2021 Key Project of Natural Science and Technology of Yangzhou Polytechnic Institute, Active Disturbance Rejection and Fault-Tolerant Control of Multi-Rotor Plant Protection UAV Based on QBall-X4 (Grant Number 2021xjzk002). 2023 Jiangsu Association for Science and Technology Youth Science and Technology Talent Promotion Project Funded Training Project. Yangzhou City School Cooperation Project "Research on Fault Tolerant Control of Multi Rotor Plant Protection UAV Based on Active disturbance Rejection Technology".

Author Contributions: The authors confirm contributions to the paper as follows: study conception and design: Lianghao Hua. Jianfeng Zhang; data collection: Dejie Li; analysis and interpretation of results: Lianghao Hua, Jianfeng Zhang, Xiaobo Xi; draft manuscript preparation: Jianfeng Zhang, Xiaobo Xi. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Hafeez, A., Husain, M. A., Singh, S. P., Chauhan, A., Mohd, T. K. et al. (2022). Implementation of drone technology for farm monitoring & pesticide spraying: A review. *Information Processing in Agriculture, 10(2),* 192–203.
2. Vargas-Ramírez, N., Paneque-Gálvez, J. (2019). The global emergence of community drones (2012–2017). *Drones, 3,* 1–24. https://doi.org/10.3390/drones3040076
3. Gayathri Devi, K., Sowmiya, N., Yasoda, K., Muthulakshmi, K., Kishore, B. (2020). Review on application of drones for crop health monitoring and spraying pesticides and fertilizer. *Journal of Critical Reviews, 7,* 667–672. https://doi.org/10.31838/jcr.07.06.117
4. Hassler, S. C., Baisal-Gurrel, F. (2019). Unmanned aerial system (UAS) in agriculture. *Agronomy, 9(10),* 618. https://doi.org/10.3390/agronomy9100618
5. San Juan, V., Santos, M., Andújar, J. M. (2018). Intelligent UAV map generation and discrete path planning for search and rescue operations. *Complexity, 2018,* 1–17. https://doi.org/10.1155/2018/6879419
6. Wang, Z. G., Zhao, H., Duan, D. G., Jiao, Y. Y., Li, J. B. (2020). Application of improved active disturbance rejection control algorithm in tilt quad rotor. *Chinese Journal of Aeronautics, 33(6),* 1625–1641. https://doi.org/10.1016/j.cja.2020.01.002
7. Sierra-García, J. E., Santos, M. (2021). Intelligent control of an UAV with a cable-suspended load using a neural network estimator. *Expert Systems with Applications, 183,* 115380. https://doi.org/10.1016/j.eswa.2021.115380
8. Ameya, R. G., Kamesh, S. (2019). Nonlinear control of unmanned aerial vehicles with cable suspended payloads. *Aerospace Science and Technology, 93,* 105299. https://doi.org/10.1016/j.ast.2019.07.032
9. Jasim, O. A., Veres, S. M. (2020). A robust controller for multi rotor UAVs. *Aerospace Science and Technology, 105,* 1270–9638. https://doi.org/10.1016/j.ast.2020.106010
10. Park, S., Lee, S., Im, B., Lee, D., Shin, S. (2023). Improvement of a multi-rotor UAV flight response simulation influenced by gust. *Aerospace Science and Technology, 134,* 1270–9638. https://doi.org/10.1016/j.ast.2023.108156
11. Papachristos, C., Alexis, K., Tzes, A. (2011). Design and experimental attitude control of an unmanned tilt-rotor aerial vehicle. *IEEE 15th International Conference on Advanced Robotics: New Boundaries for Robotics,* 465–470. https://doi.org/10.1109/icar.2011.6088631

12. Merheb, A. R., Noura, H., Bateman, F. (2014). Active fault tolerant control of quadrotor UAV using sliding mode control. *International Conference on Unmanned Aircraft Systems,* 156–166. https://doi.org/10.1109/icuas.2014.6842251
13. Wen, H., Liang, Z., Zhou, H., Li, X., Yao, B. et al. (2023). Adaptive sliding mode control for unknown uncertain non-linear systems with variable coefficients and disturbances. *Communications in Nonlinear Science and Numerical Simulation, 121,* 107225. https://doi.org/10.1016/j.cnsns.2023.107225
14. Xu, W. H., Cao, L. J., Peng, B. Y., Wang, L., Gen, C. et al. (2023). Adaptive nonsingular fast terminal sliding mode control of aerial manipulation based on nonlinear disturbance observer. *Drones, 7(2),* 88. https://doi.org/10.3390/drones7020088
15. Zhou, L. S., Ma, L. L., Wang, J. Z. (2017). Fault tolerant control for a class of nonlinear system based on active disturbance rejection control and RBF neural networks. *Chinese Control Conference*, pp. 7321–7326. Dalian, China. https://doi.org/10.23919/ChiCC.2017.8028513
16. Zhong, Y. J., Liu, Z. X., Zhang, Y. M., Zhang, W., Zuo, J. Y. (2019). Active fault-tolerant tracking control of a quadrotor with model uncertainties and actuator faults. *Frontiers of Information Technology & Electronic Engineering, 20(1),* 95–106. https://doi.org/10.1631/FITEE.1800570
17. Wang, X., Sun, H., Long, Y. W., Zheng, L. H., Liu, H. J. et al. (2018). Development of visualization system for agricultural UAV crop growth information collection. *IFAC-PapersOnLine, 51(17),* 631–636. https://doi.org/10.1016/j.ifacol.2018.08.126
18. Antonella, B., Marko, C., Stjepan, B. (2019). Vision-based system for a real-time detection and following of UAV. arXiv:2205.00083.
19. Torsten, F., Hans-Jurgen, S., Rudiger, S. (2019). Advances in the development of an industrial UAV for large-scale near-field antenna measurements. *13th European Conference on Antennas and Propagation (EuCAP)*, pp. 1–3. Krakow, Poland.
20. Ding, M. L., Tang, L., Zhou, L. J., Wang, X. M., Weng, Z. L. et al. (2019). W band mini-SAR on multi rotor UAV platform. *Proceedings of 2019 IEEE 2nd International Conference on Electronic Information and Communication Technology*, pp. 416–418. Harbin, China. https://doi.org/10.1109/iceict.2019.8846356
21. Lu, H. C., Chang, M. H., Tsai, C. H. (2011). Adaptive self-constructing fuzzy neural network controller for hardware implementation of an inverted pendulum system. *Applied Soft Computing, 11(5),* 3962–3975. https://doi.org/10.1016/j.asoc.2011.02.025
22. Liu, L., Liu, Y., Zhou, L., Wang, B., Cheng, Z. et al. (2022). Cascade ADRC with neural network-based ESO for hypersonic vehicle. *Journal of the Franklin Institute, 360(12),* 9115–9138. https://doi.org/10.1016/j.jfranklin.2022.09.019
23. Yu, X., Fu, Y., Li, P., Zhang, Y. M. (2018). Fault-tolerant aircraft control based on self-constructing fuzzy neural networks and multivariable SMC under actuator faults. *IEEE Transactions on Fuzzy Systems, 26(4),* 2324–2335. https://doi.org/10.1109/tfuzz.2017.2773422
24. Li, T., Zhang, Y. M., Gordon, B. W. (2012). Nonlinear fault-tolerant control of a quadrotor UAV based on sliding mode control technique. *IFAC Proceedings Volumes, 8(1),* 1317–1322. https://doi.org/10.3182/20120829-3-mx-2028.00056
25. Wang, J., Zhao, Z. H., Zheng, Y. (2018). NFTSM-based fault tolerant control for quadrotor unmanned aerial vehicle with finite-time convergence. *IFAC-PapersOnLine, 51(24),* 441–446. https://doi.org/10.1016/j.ifacol.2018.09.614
26. Amoozgar, M. H., Chamseddine, A., Zhang, Y. M. (2012). Fault-tolerant fuzzy gain-scheduled PID for a quadrotor helicopter testbed in the presence of actuator faults. *IFAC Proceedings Volumes, 2(1),* 282–2870. https://doi.org/10.3182/20120328-3-it-3014.00048
27. Boche, A., Farges, J. L., de Plinval, H. (2017). Reconfiguration control method for multiple actuator faults on UAV. *IFAC-PapersOnLine, 50(1),* 12691–12697. https://doi.org/10.1016/j.ifacol.2017.08.2257

28. Wen, S. P., Chen, M. Z. Q., Zeng, Z. G., Huang, T. W., Li, C. J. (2017). Adaptive neural-fuzzy sliding-mode fault-tolerant control for uncertain nonlinear systems. *IEEE Transactions on Systems Man Cybernetics-Systems, 47(8),* 2268–2278. https://doi.org/10.1109/tsmc.2017.2648826
29. Xiang, Y., Zhang, Y. M. (2015). Design of passive fault-tolerant flight controller against actuator failures. *Chinese Journal of Aeronautics, 28(1),* 180–190. https://doi.org/10.1016/j.cja.2014.12.006
30. Ai, S., Song, J., Cai, G., Zhao, K. (2022). Active fault-tolerant control for quadrotor UAV against sensor fault diagnosed by the auto sequential random forest. *Aerospace, 9,* 518. https://doi.org/10.3390/aerospace9090518
31. Hua, L. H., Zhang, J. F., Li, D. J., Xi, X. B. (2021). Fault-tolerant active disturbance rejection control of plant protection of unmanned aerial vehicles based on a spatio-temporal RBF neural network. *Applied Sciences, 11(9),* 4084. https://doi.org/10.3390/app11094084
32. Wang, S. B., Han, Y., Chen, J., Du, N., Pan, Y. et al. (2018). Flight safety strategy analysis of the plant protection UAV. *IFAC-PapersOnLine, 51(17),* 262–267. https://doi.org/10.1016/j.ifacol.2018.08.170
33. Guerrero-Sánchez, M. E., Hernández-González, O., Valencia-Palomo, G., Mercado-Ravell, D. A., López-Estrada, F. R. et al. (2021). Robust IDA-PBC for under-actuated systems with inertia matrix dependent of the unactuated coordinates: Application to a UAV carrying a load. *Nonlinear Dynamics, 105,* 3225–3238. https://doi.org/10.1007/s11071-021-06776-7
34. Guerrero-Sanchez, M. E., Lozano, R., Castillo, P., Hernandez-Gonzalez, O., Garcia-Beltran, C. D. et al. (2021). Nonlinear control strategies for a UAV carrying a load with swing attenuation. *Applied Mathematical Modelling, 91,* 709–722. https://doi.org/10.1016/j.apm.2020.09.027
35. Herbst, G. (2013). A simulative study on active disturbance rejection control (ADRC) as a control tool for practitioners. *Electronics, 2(3),* 246–279. https://doi.org/10.3390/electronics2030246
36. Haykin, S. O. (1994). *Neural networks: A comprehensive foundation*. Upper Saddle River: Prentice Hall PTR.
37. Aftab, W., Moinuddin, M., Shaikh, M. S. (2014). A novel kernel for RBF based neural networks. *Abstract and Applied Analysis, 2014,* 1–10. https://doi.org/10.1155/2014/176253